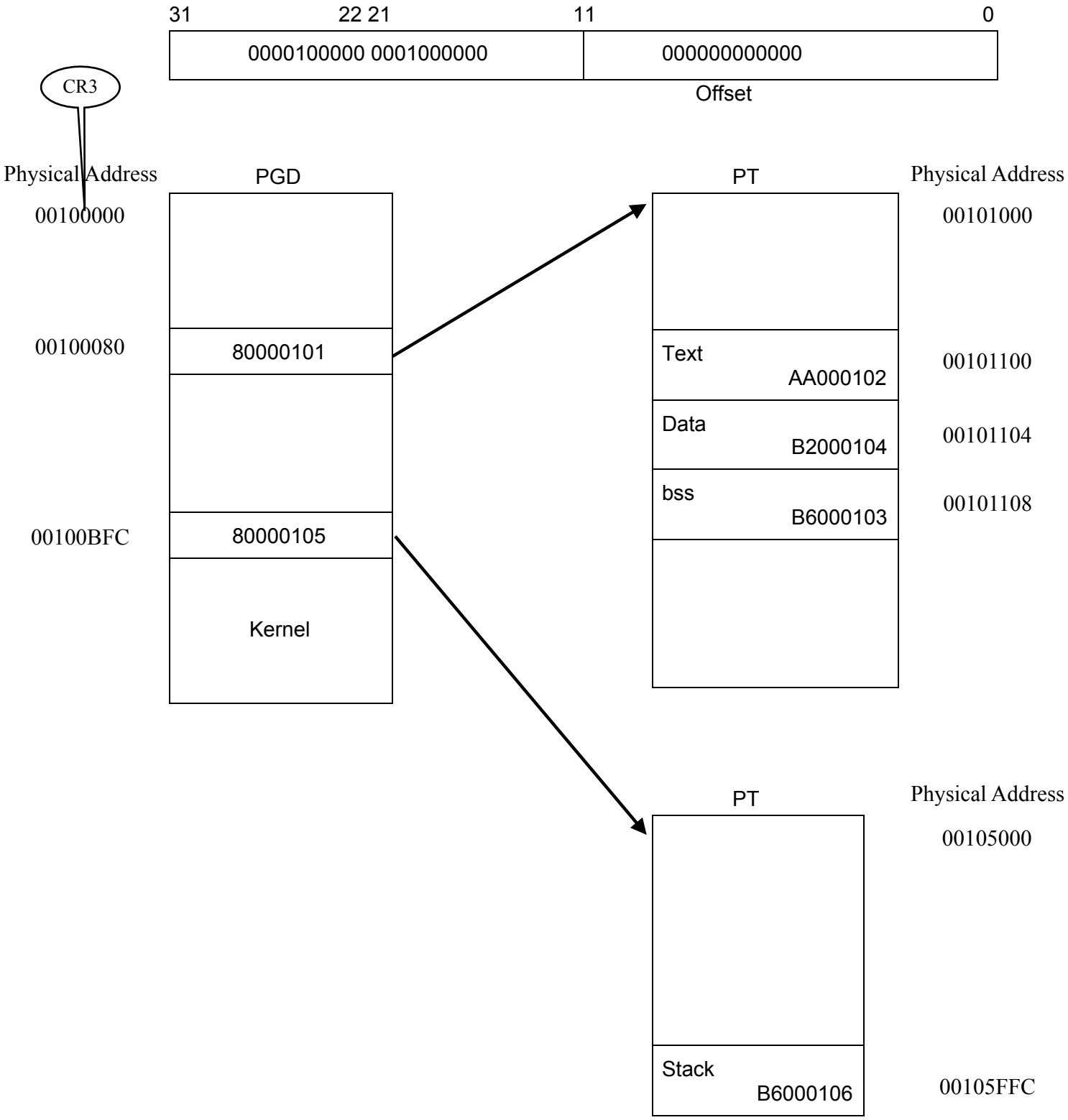
1.

Virtual address: 0x08040000

Binary representation: 0000100000 0001000000 000000000000



2.

a. To achieve strict LRU, we would need to keep track of the page with the longest idle time. But it is extremely difficult to achieve this behavior for the PFRA algorithm, the reason is that PFRA claims back pages that were already mapped without knowing the time stamp of the page. So to achieve LRU, we would need some sort of timer along with an access bit that can update the timer every time a certain file is accessed. However, we only have a bool access bit so it is not possible to achieve full replication.
   PFRA has two separate page frames, one for active and one for non-active. To find the page in the active frame to bring to inactive, the PTE is examined to check on the access bit, and clears it if it is set. Then the reference bit will keep track of whether or not the access bit was set before and if it is, then the reference bit will be up. Each scan will look at the two bits, and if the reference and the access bit are 0, then the page will be moved to inactive.

b. An address_space structure can help to look up and locate the page frame that contains a certain offset in a file in a timely manner. One type of tree, which is a radix-64 tree will find the part of the file given the offset, or it will return NULL if it does not exist. This data structure is used for paging-in operation if a minor fault can be resolved by pointing the PTE at the existing page frame which contains the correct image.

c. When handling a page fault, where the faulting VA is below the user/kernel line, we would access the vm_area_struct which can tell us whether or not SIGSEGV should be delivered. This is done by examining the VA space descriptor to see if the file is within the range, then we would check the protection violation. If the address is outside of a valid region or protection violation occurred, a SIGSEGV is posted to the process.